

SW ENGINEERING

CSC648/848 Fall 2019

Section: 01 | Team #07
Global Team

Roderic Kong - Team Lead
Email: rkong1@mail.sfsu.edu

Chris Jol - Front End Lead
Kevin Vanagas - Back End Lead
Deep Bueller - Git Master

Milestone 2

10/09/2019

Date
10/12/2019

Submission
01

Revisions
0

1. Functional Requirements

Priority 1

Unregistered Users

1. User shall be able to make an account
2. User shall be able to search for posts
3. Users shall be able to filter by category
4. Users shall not be prompted to make an account until they perform an action that requires one
5. Users Shall have access to nav bar on every page

Registered Users

6. Users shall be able to log in
7. Users shall be able to create a post
8. Users Shall not be prompted to log in upon loading the site
9. Users shall not be redirected away from their current page in order to log in

Admins

10. Admins shall be required to approve posts before going live
11. Admins shall have access to delete live posts
12. Admins shall not have access to edit posts

Priority 2

Unregistered Users

13. Users shall have access recent and featured posts from the homepage
14. User's search query shall be maintained in search bar until changed by user

Registered Users

15. Users shall be able to send a message to the creator of a post
16. Users shall be able to edit their posts after its initial submission

Priority 3

Registered Users

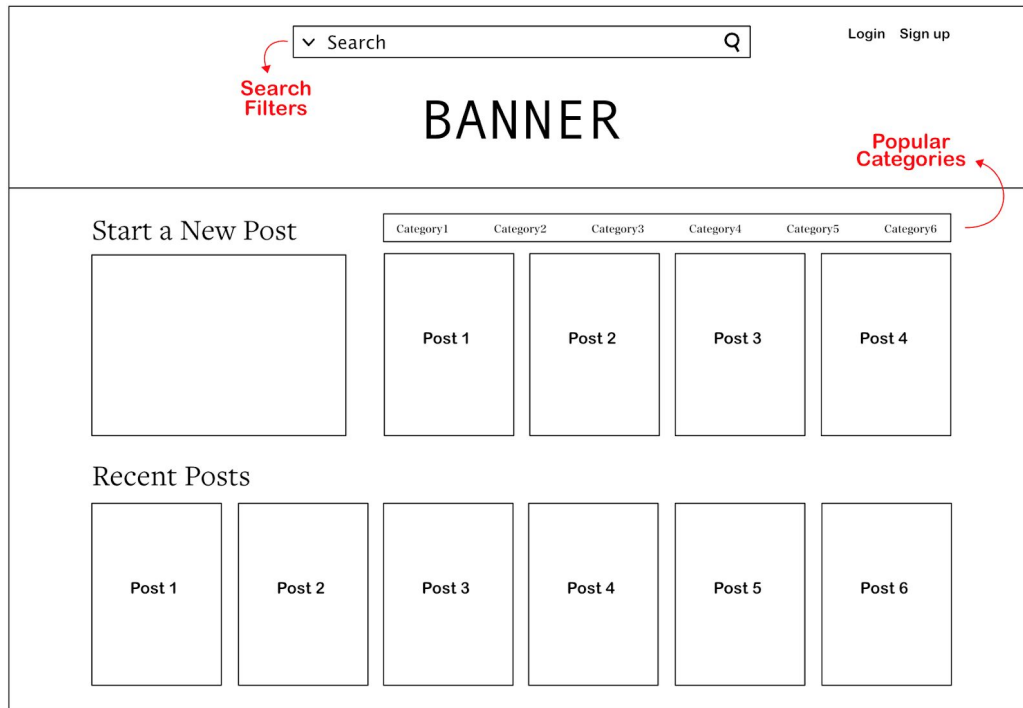
- 17. Users shall be able to tag their posts to help increase search hits
- 18. Users shall be able to comment on and like posts
- 19. Likes and Comments shall increase post popularity rating

Admins

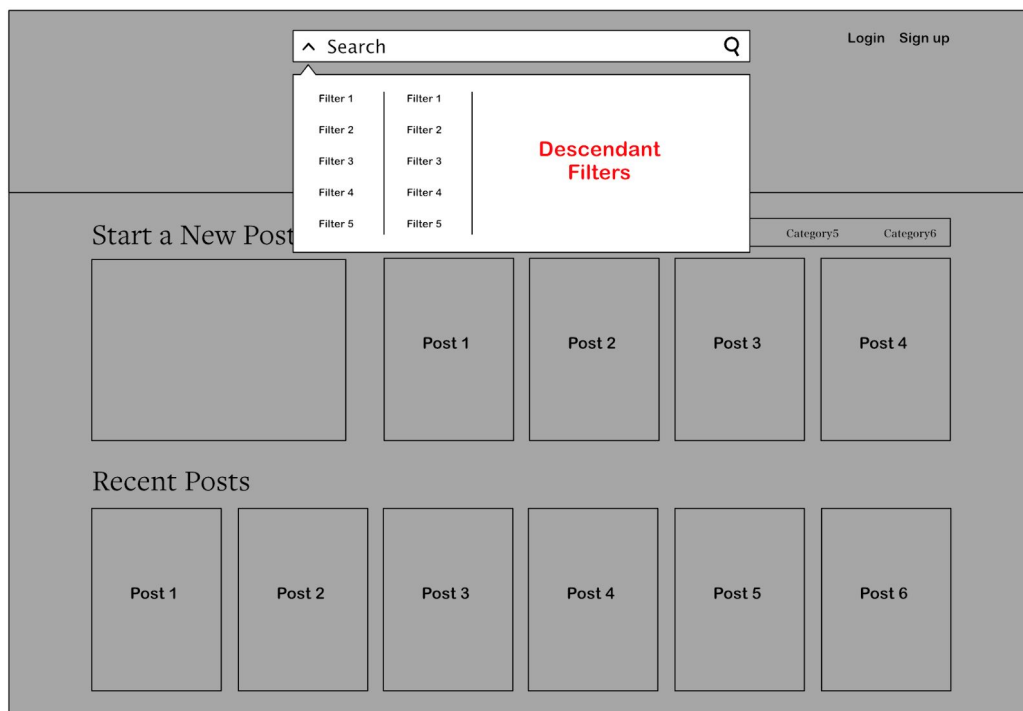
- 20. Admins shall have the ability to give feedback for post to meet qualifications

2. UI Mockups & Storyboards

Upon loading the site Asana is delighted to see that it's full of posts from other students both requesting and posting information/resources.



She quickly sees that she can filter the posts by her major.



She finds that there are already posts from people selling their old textbooks for much cheaper than what the bookstore offers. And what do you know, that same classmate just made a post asking if anyone wants to meetup to study

The wireframe shows a web application layout. At the top right, there are links for "Login" and "Sign up". Below these is a search bar with a placeholder text "Search" and a magnifying glass icon. The main content area is divided into two sections. On the left is a vertical sidebar labeled "Filters". On the right is a list of four posts, each represented by a rectangular box labeled "Post 1", "Post 2", "Post 3", and "Post 4" respectively.

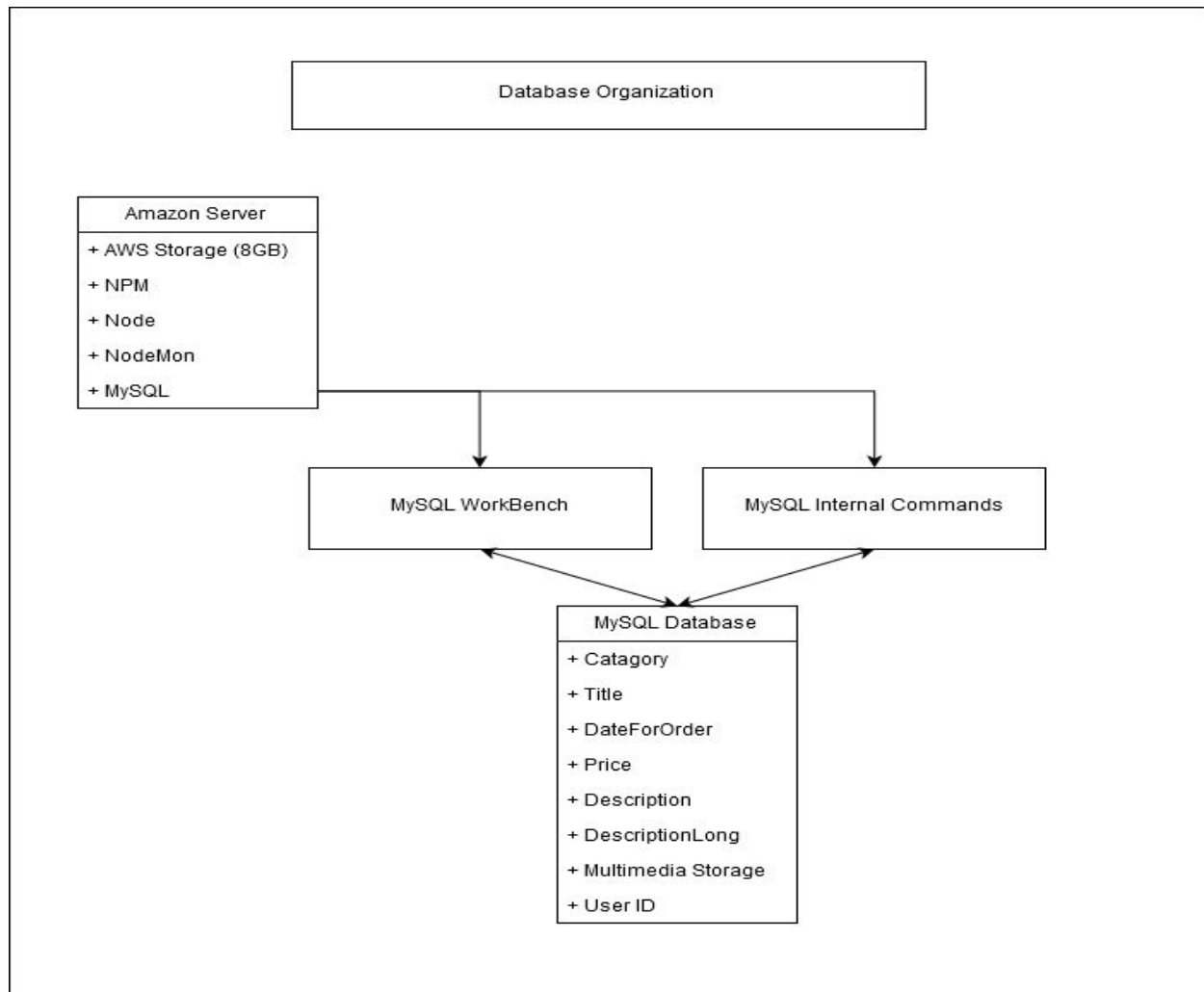
Bob logs in and creates a post to sell his book where he enter information about the product, which class is using sold textbook and then submits it.

The wireframe shows a web application layout with a "New Post" modal form overlaid. The modal form has a title "New Post" and contains the following fields: "Name:" with a text input, "Description:" with a text area, and "Category:" with a dropdown menu showing "CompSci". The background of the page is a light gray and contains several UI elements: a search bar at the top with a placeholder "Search" and a magnifying glass icon; links for "Login" and "Sign up" at the top right; a "Start a New" button on the left; a "Recent Post" section on the left with a box labeled "Post 1"; a grid of six post boxes in the center labeled "Post 2", "Post 3", "Post 4", "Post 5", and "Post 6"; and a "Category5" and "Category6" dropdown menu on the right.

3. High-level Architecture, Database Organization

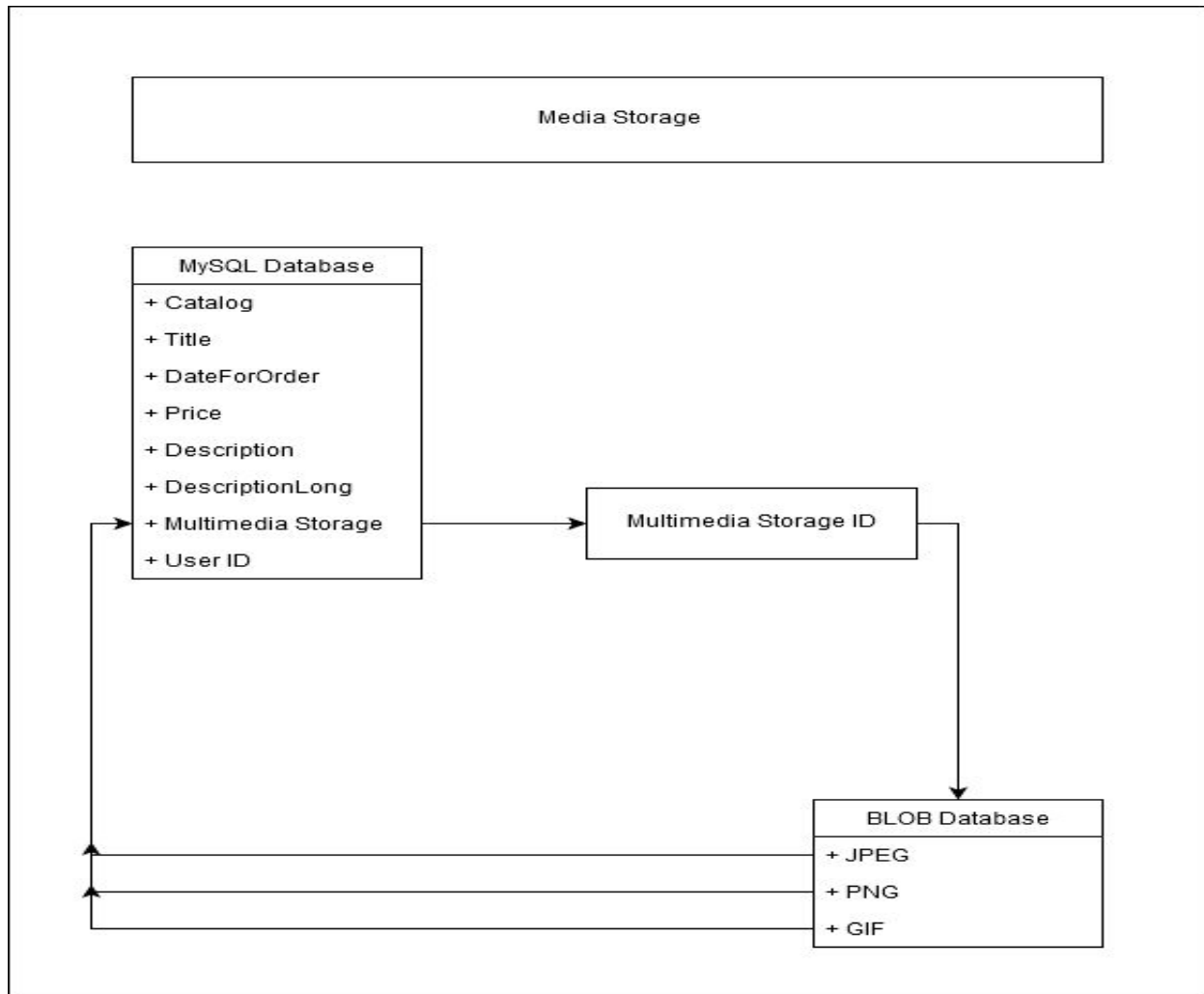
1. Database Organization

- a. Within Amazon's Server, database will be managed by MySQL
- b. MySQL will be managed by either it's internal commands or WorkBench.
- c. MySQL Database will consist of a table with listed categories.



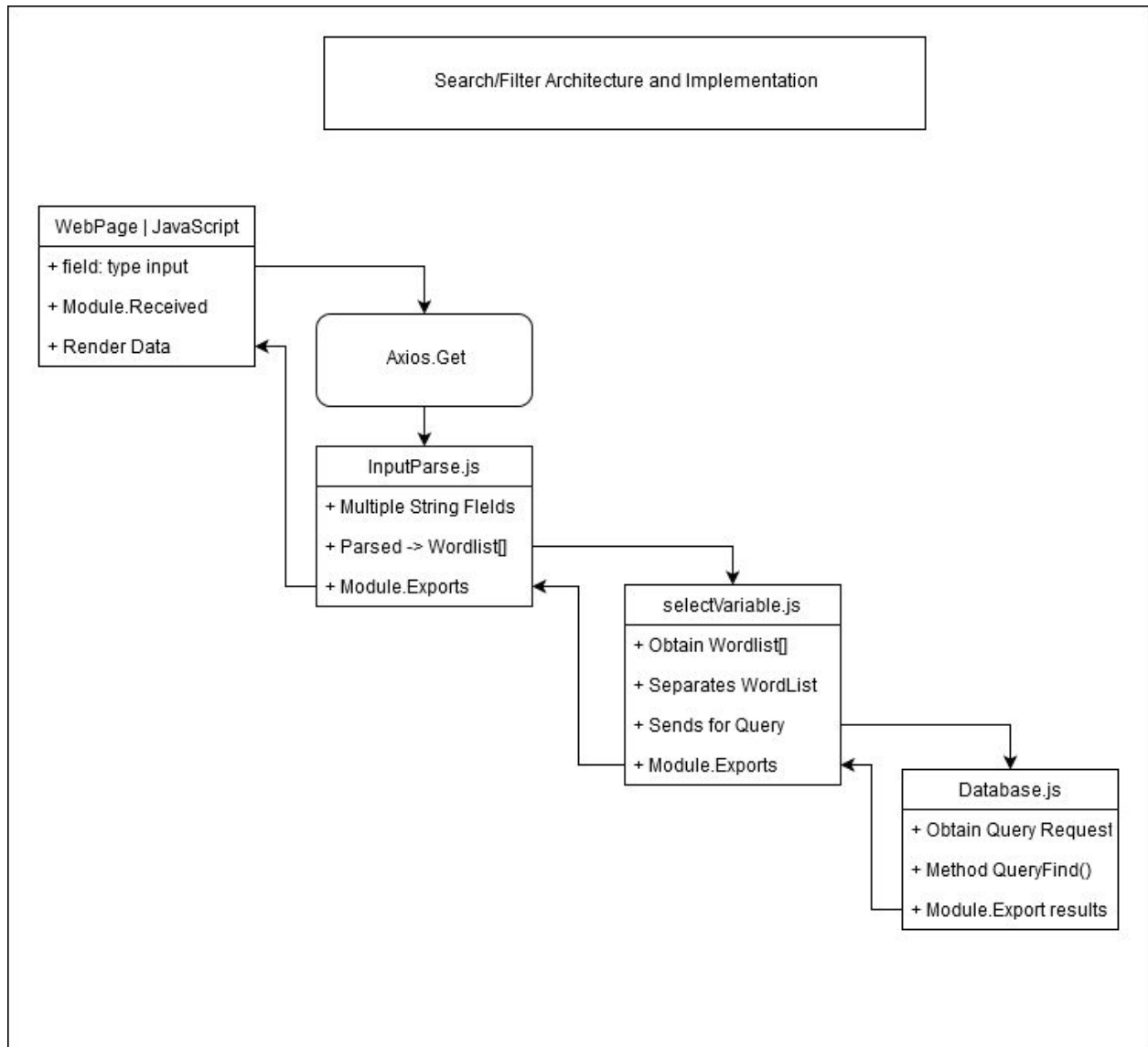
- i. Category - Type of item selling.
- ii. Title - Name of Item.
- iii. DateForOrder - Post date.
- iv. Price - Item Selling price.
- v. Description - Description for Thumbnail.
- vi. DescriptionLong- Description for more intricate Post
- vii. MultiMedia - Storage embedded within MySQL Table
- viii. User ID - Who requested to post this item

Media Storage.



1. In MySQL, Blob is used to store media files such as photos.
2. When retrieving image files. `SELECT * FROM [TABLE] WHERE [Multimedia Column]`
3. The photo will be then sent to be posted on MySQL table to be rendered with our JS file

Search/Filter Architecture and Implementation.



1. When user inputs are selected from the search bar, the inputs are obtained via Axios' internal function 'Get'.
2. The complete string will be grabbed from the search bar, then used in our InputParse.js.
3. In our InputParse.js we will parse our string into multiple verses, and then append each word into an array.
4. We will then send the array to SelectVariable.js, which will send for query in tandem of each array word.
5. Once it is queried with MySQL, it is then sent back with recursion into the front WebPage to be displayed

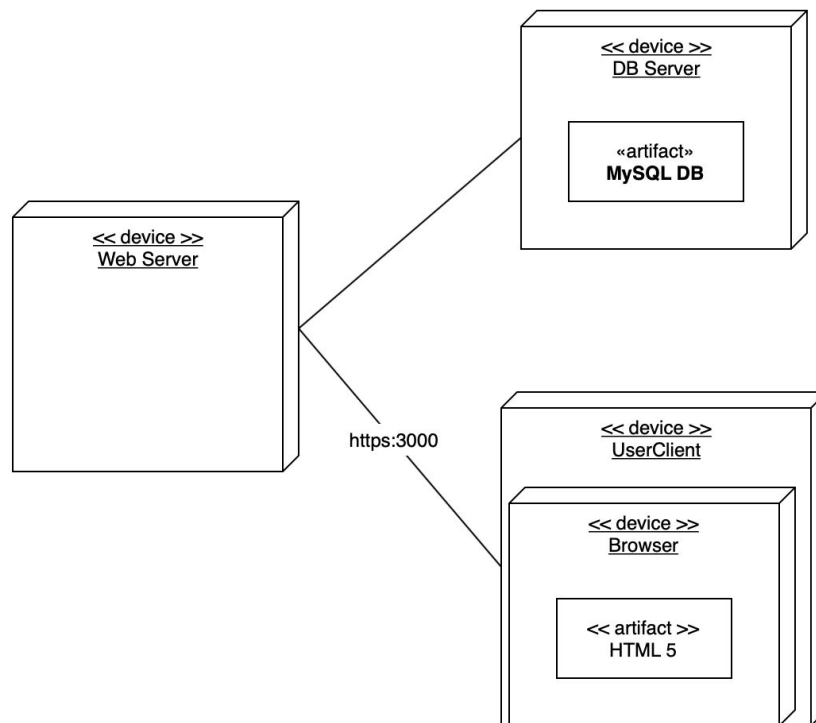
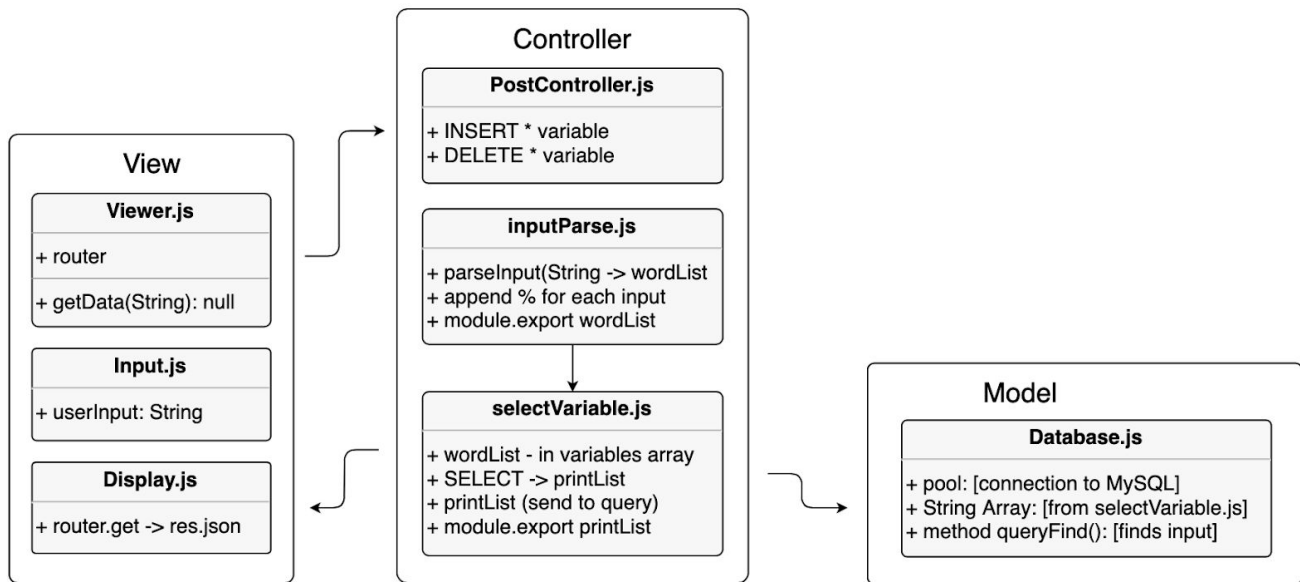
API

1. Axios - A bridgework between React Javascript and Express JS

SEARCHING ALGORITHMS | PROCESSES

1. Searching Algorithms.
 - a. As for the basis of this current Milestone, we are emphasizing functionalities along with a small database.
 - b. Linear Search will be currently used. Despite the linear slowness of the searching algorithm, we would like to focus on getting our code to work first before implementing any optimization.
 - c. After successfully implementation of codes, we will experiment with other potentially faster searching algorithm.
2. Processes.
 - a. Sorting functionalities such as 'like rating', 'ranking', 'sort by price' will come later in development. However as for our planned process, here are the chronological order of sorting options:
 - i. Sort by Category.
 - ii. Sort by Latest.
 - iii. Sort by Highest to Lowest Price (General).
 - iv. Sort by Lowest to Highest Price (General).
 - v. Sort by Highest to Lowest Price (Category).
 - vi. Sort by Lowest to Highest Price (Category).

4. High-level UML Diagrams



5. Key Risks

1. Skill Risks:
 - a. Team members have limited experience with MySQL.
 - b. Some team members have limited experience with working with Web Responses and Requests.
 - c. Some team members may have limited experience with Axios.
2. Schedule Risks:
 - a. As our team is working on a full development of a Webapp within 2 months, along with other coursework priorities, there might be some functions commonly found in professional selling webapps that may be triaged and potentially unfulfilled.
3. Technical Risks:
 - a. Structuring the front-end in a way so that it make as few database requests as possible. Too many redundant requests can cost a lot of money. Need to find a way to cache relevant data that is maintained throughout a user session.
 - b. There might be some potential for conflict between API, and/or middleware.
4. Teamwork Risks:
 - a. As a global team communication between our team and the team from Germany could be a potential risk due to time zone conflicts.
 - b. As more midterm exams and projects are due, time spent on this project will need to be split between other projects. Many meet-ups may miss some members.
 - c. Lack of organization between team members.
5. Legal/Content Risks:
 - a. Many of our researched middleware are listed as completely free to use. Since no money value will be extracted utilizing State Exchange, legal risk of copyright or demands of any compensation will be nil.
 - b. Our only risk will be that demo photography used to demonstrate multimedia retrieval may be copyrighted. However, since it is used in a school environment, any legal risk of photos obtained online or photographed ourselves will be minimal.

Risk resolution:

1. Skill Risk resolution:
 - a. Team 07 went from zero knowledge of any back-end development, to creating MySQL databases, scripts to access MySQL databases, within 3 weeks. Our team will devote itself to learning the required tools to creating the WebApp.
2. Schedule Risk resolution:
 - a. With our current schedule, a full international team meet-up may not be possible. Intended resolution will be using Zoom or other free video call software, set up a chatroom, and specify a time that is reasonable for global members.
3. Technical Risk resolution:
 - a. Extensive QA done by designated member will be done and reports about errors or bugs will be made to all to work on.
4. Teammate Risk resolution:
 - a. For the American team spearheading this project, we will be creating direct paths and instructions for the German Team, to minimize confusion and to streamline workflow. To accomplish this, a MVC framework which will allow for easier modular expansion will be created, therefore any implementation from the German Team will be swift and seamless.
 - b. Trillo - an organizational tool introduced by Chris is our method of organization. Trillo will tackle our lists to do for each portion of the project and allows us to set due dates, allowing us to track our progress.
5. Legal/Content Risk resolution:
 - a. Any content (aside from SFSU banner, and SFSU logo) will be created using personal devices and skills. Images and content will be based from either free stock photos, or our personal team creations.

6. Project Management

For this project, we have both the back and front end work in tandem for the project. In order for the project to proceed without any hitch, our team as already agreed on working with our current framework. We plan on completing the beginning of the WebApp frame-work following a MVC software design. By following the MVC design, we have accomplished connection between Back End and MySQL, which is controlled by “database.js” within the Model Folder. Webpage GUI is currently a basic homepage that fulfilled user input, and displaying information from MySQL controlled by “viewer.js” inside the View folder.

Our future tasks will be deeper implementation of queries within our Controller folders, which will allow parsing of search bar inputs, filtering search results based on parsed inputs. Another task we will have to explore will be multimedia storing. Our discussion will be either storing the picture and video files into a MySQL database as BLOBs’ or storing it into another distinct folder and utilizing pathname retrieval as a method of access. After implementation of multimedia storage, we will also work with account registrations with password encryptions, and user interfaces for Administrators.